

Modellierung von On-Chip-Trace-Architekturen für eingebettete Systeme

Kai-Uwe Irrgang

Dresden, 05.06.2015



DRESDEN
concept
Exzellenz aus
Wissenschaft
und Kultur

Gliederung

1. Einleitung

2. Stand der Technik

3. Modellierung der On-Chip-Trace-Architektur

4. Ergebnisse

5. Zusammenfassung

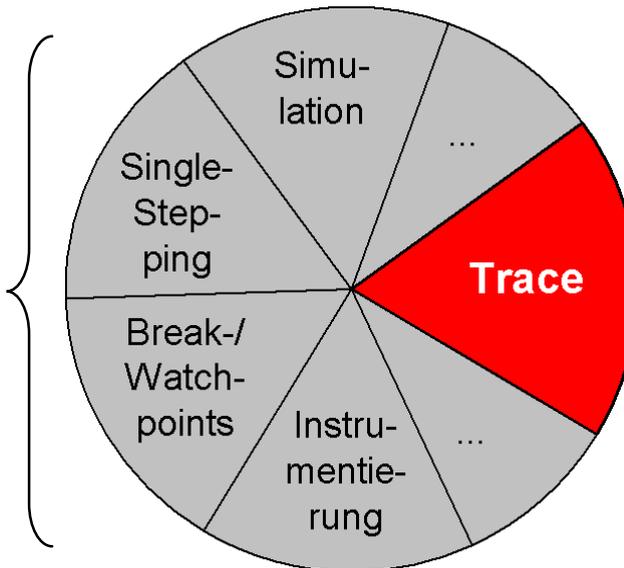
Einleitung

Eingebettete Systeme

- zahlreiche Anwendungsgebiete: Automobiltechnik, Medizintechnik, ...
- teilweise sehr hohe **Sicherheitsanforderungen** und/oder harte **Echtzeitanforderungen** (z. B. Herzschrittmacher, Bremssystem, ...)
- enthalten typischerweise Prozessoren und damit **Software**

=> **Software-Test**

- nicht echtzeitfähig
- Maskierung von Fehlern möglich

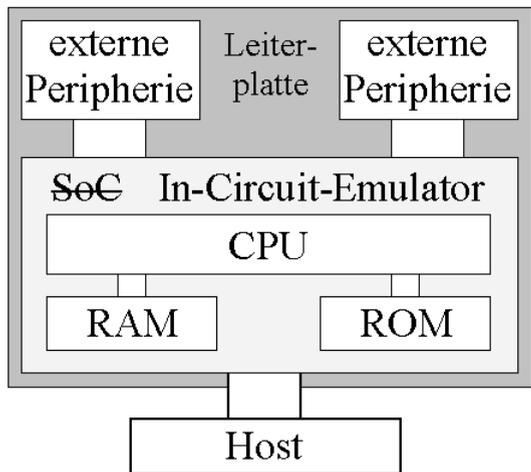


= wesentlicher Bestandteil des Software-Tests

= rückwirkungsfreies Aufzeichnen von Systemzuständen
... in Echtzeit, während das System in Interaktion mit der Systemumgebung läuft

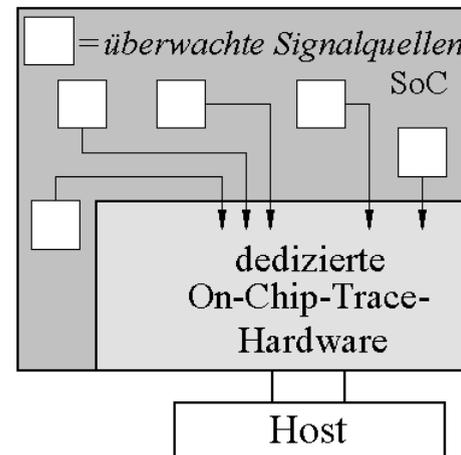
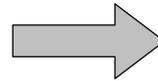
Einleitung

Übergang vom Off-Chip-Trace zum On-Chip-Trace



Off-Chip-Trace

- In-Circuit-Emulator und Logikanalysator
- + sehr leistungsfähig
- stetig sinkende Einsetzbarkeit

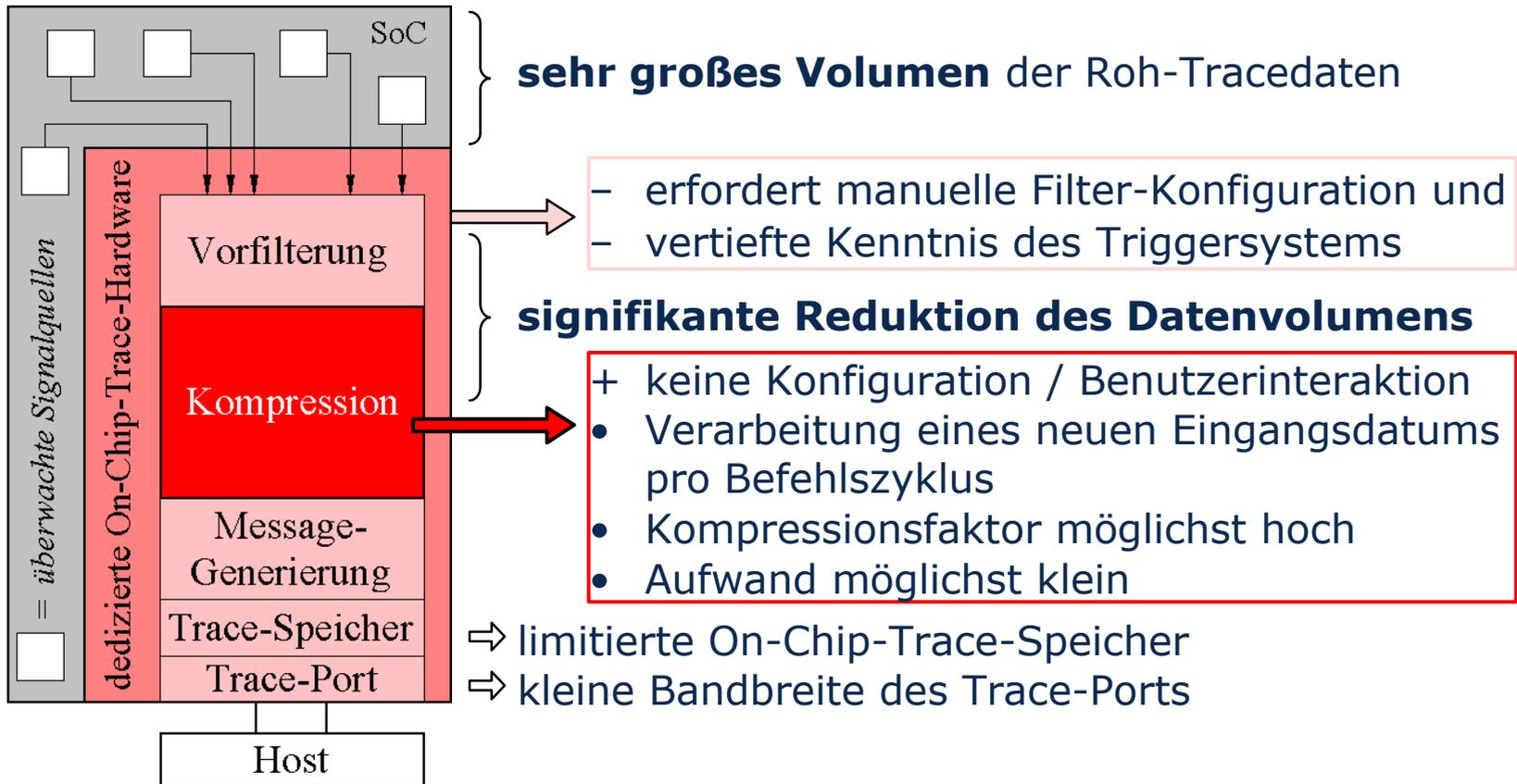


On-Chip-Trace: Integration dedizierter On-Chip-Trace-Hardware

- Aufwand muss klein sein
- Universalität vor Spezialisierung
- + skaliert mit der Halbleitertechnologie
- => gleiche Taktfrequenzen realisierbar

Einleitung

On-Chip-Trace - Herausforderungen und Ziele



Gliederung

1. Einleitung

2. Stand der Technik

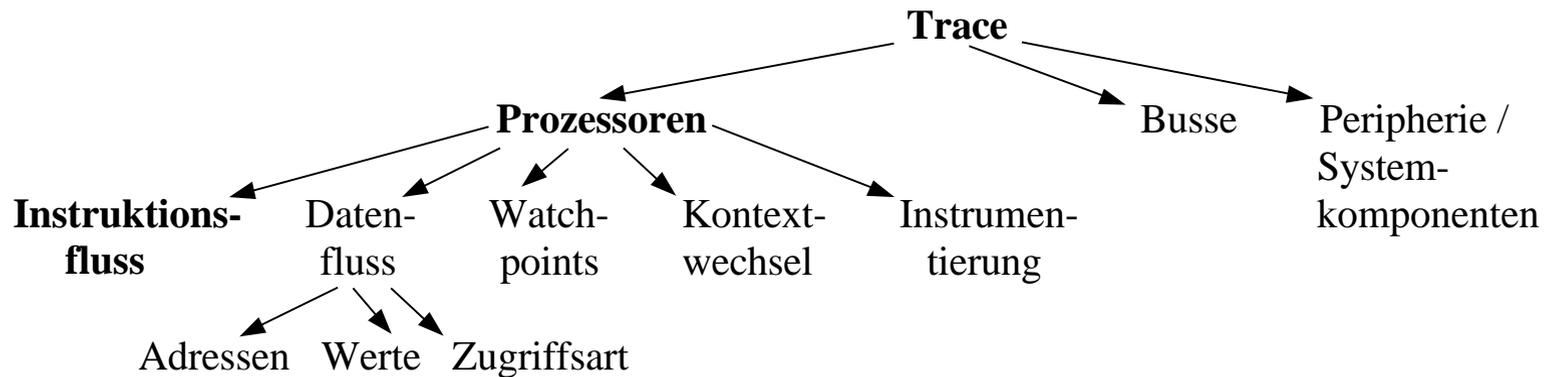
3. Modellierung der On-Chip-Trace-Architektur

4. Ergebnisse

5. Zusammenfassung

Stand der Technik

Quellen und Arten von Tracedaten



- ein **vollständiger Instruktionstrace** ist am wichtigsten
- andere Tracedaten sind oft nur in dessen Kontext bewertbar

Stand der Technik

Verlustlose Kompression

- Huffman-Codierung und arithmetische Codierung
 - Kenntnis der Quellenstatistik notwendig
 - => Ermittlung erfordert zusätzlichen Durchlauf durch die Daten
 - => für Echtzeit-Verarbeitung eher ungeeignet
 - adaptive Varianten sind zu aufwendig für Hardware-Implementierung
- **wörterbuchbasierte Kompression**
 - + **adaptive Anpassung** an die Quellenstatistik
 - hoher Aufwand für das Speichern des Wörterbuches
- Vorverarbeitung der Daten und Kombination der Verfahren
 - bzip2: Burrows-Wheeler-Transformation + **Move-to-Front** + Huffman
 - Deflate (gzip): LZSS + Huffman
 - ...
- **Program-Flow-Change-Model (PFCM)**: Redundanzreduktion auf der Basis des statischen Programmcodes beim Instruktionstrace

Stand der Technik

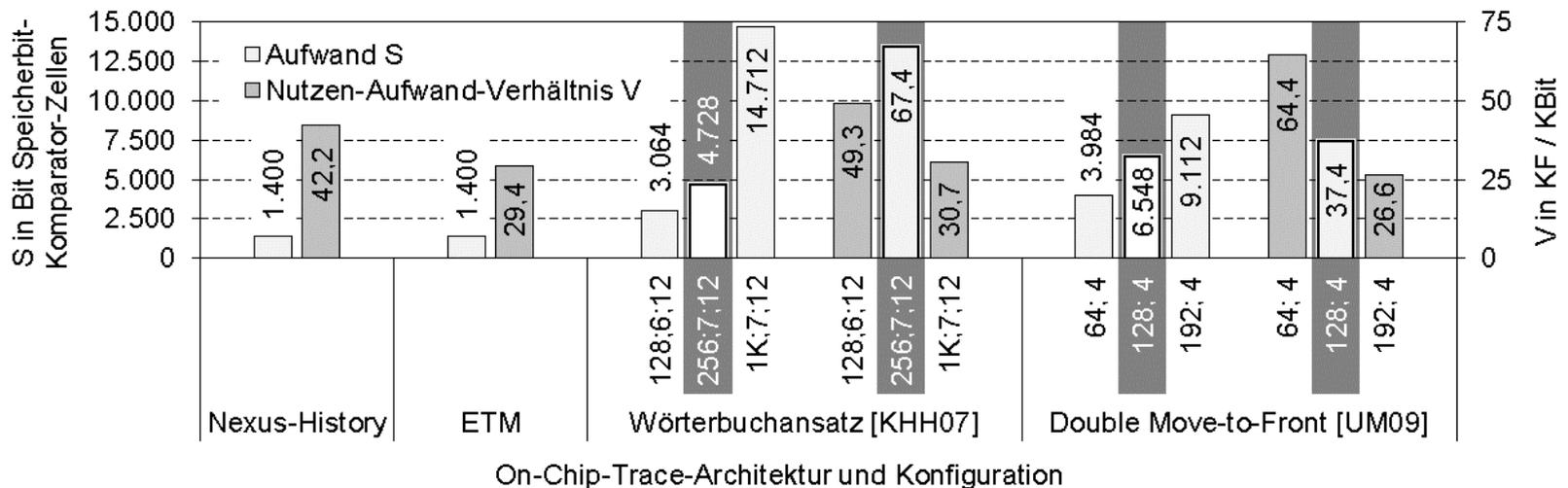
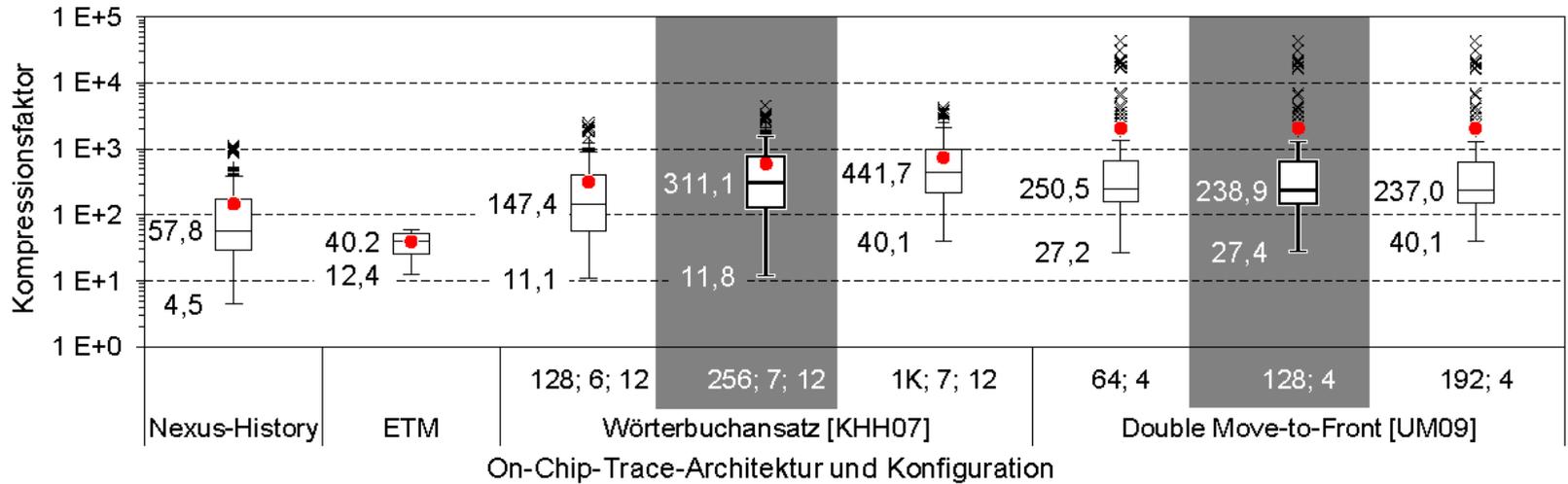
Merkmale ausgewählter On-Chip-Trace-Architekturen

Merkmale		Nexus IEEE-ISTO 5001 [IEE03]	ARM CoreSight ETM [ARM13b]	Wörterbuchansatz von Kao et al. [KHH07]	Double-Move-to-Front (DMTF) von Uzelac et al. [UM09]
Tracearten	Instruktionsfluss	ja	ja	ja	ja
	Datenfluss	optional	optional	nein	nein
	Kontextwechsel	optional	optional	nein	nein
	Watchpoints	optional	optional	nein	nein
Daten-redukt.	Relevanzfilterung	nein	optional	nein	nein
	Kompression	PFCM + XOR auf Adressen	PFCM + XOR auf Adressen	Wörterbuch für Basisblöcke	DMTF für Instruktionsströme
Messages	Messageformat	paralleler Port, TCODEs	parallel, byteorientiert	parallele Ausgabe wie LZ77	fortlaufender Bitstrom
	Synchronisation	ja	ja	nein	nein
	Überlaufbehandlung	nein	ja	nein	nein
	Quellenidentifikation	ja	ja	nein	nein
	Multi-Core-Fähigkeit	ja	ja	nein	nein

Stand der Technik

Maßzahlen und Kriterien zur Bewertung

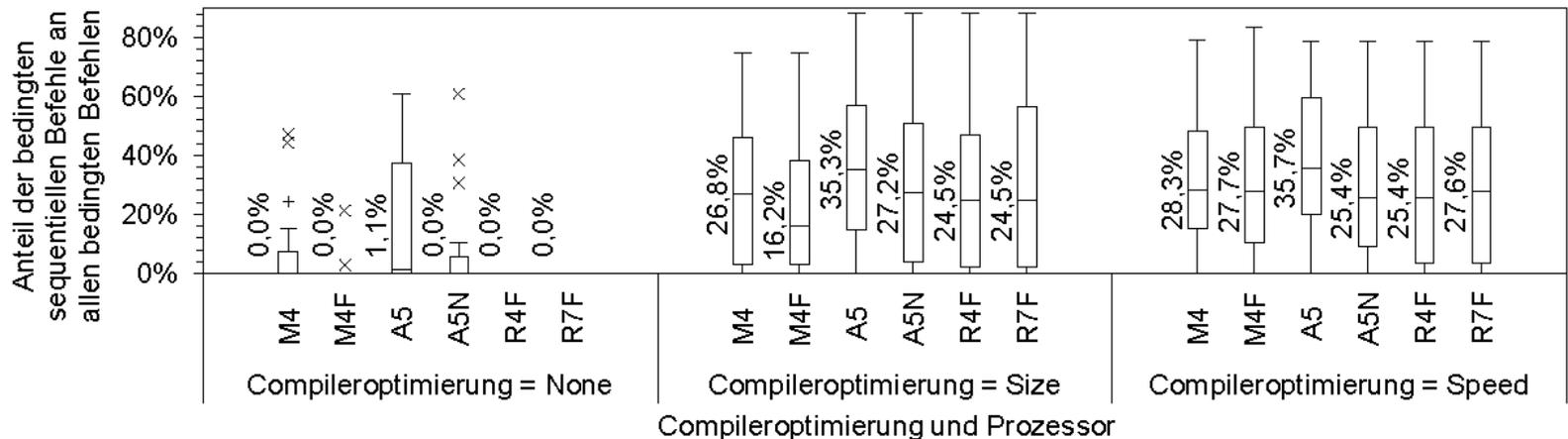
- einheitliche Untersuchungsbedingungen für vergleichende Bewertung
 - EEMBC Automotive Benchmark Software (16 Benchmark-Programme)
 - sieben ARM Cortex Prozessoren aller Familien (**M**icrocontroller, **A**pplication und **R**eal-Time), mit und ohne FPU: M0, M4, M4F, A5, A5N, R4F, R7F
 - Einbezug aller Befehlssätze (Thumb, Thumb2 und ARM32)
- **Aufwands**-Maß: Anzahl Speicherbit-Komparator-Zellen S
- **Nutzen**-Maß: Kompressionsfaktor KF
 N_{gesamt} = ausgeführte Befehle
 $B_{Ausgabe}$ = Bitmenge Ausgabedaten
$$KF = 32 \cdot \frac{N_{gesamt}}{B_{Ausgabe}} \quad (2.1)$$
- Nutzen-Aufwand-Verhältnis V
$$V = \frac{KF}{S} \quad (2.2)$$
- **Vollständigkeit** des Instruktionstrace



Stand der Technik

Vollständigkeit des Instruktionstrace - bedingte Befehle

C-Quellcode	klassische Variante	bedingte Befehle
<pre> if(a != b) { do{ if(a > b) a -= b; else b -= a; }while(a != b); } </pre>	<pre> gcd: CMP r0, r1 BEQ end BLE m01 SUBS r0, r0, r1 B gcd m01: SUBS r1, r1, r0 B gcd end: </pre>	<pre> gcd: CMP r0, r1 BEQ end m01: SUBGT r0, r0, r1 SUBLE r1, r1, r0 CMP r0, r1 BNE m01 end: </pre>



Stand der Technik

Vergleich, Verbesserungspotential und Zieldefinition

Befehlstyp		Nexus IEEE-ISTO 5001	ARM CoreSight ETM	Wörterbuchansatz Kao	Double-Move-to-Front (DMTF)	
sequen- tiell	unbedingt	zählen: I-CNT	Ausführungsbit in P-Header	Blocklänge++	Streamlänge++	
	bedingt, ausgeführt bedingt, nicht ausgef.	Ausführungsbit in History				
Verzweigung	direkt	unbedingt	Ausführungsbit in P-Header	Basisblock	Streamlänge++ Streamdeskriptor	
		bedingt, ausgeführt bedingt, nicht ausgef.				Ausführungsbit in History
	indirekt	unbedingt	Msg: Indirect-Branch-History	Msg: Branch Address	Basisblock in Kompressor	Streamdeskriptor in Kompressor
		bedingt, ausgeführt bedingt, nicht ausgef.	Ausführungsbit	Ausführungsbit	Blocklänge++	Streamlänge++

ZIEL:

- + Instruktionstrace vollständig - Instruktionstrace **UNvollständig**
 - keine/geringe Kompression + hohe Kompression
- + Instruktionstrace vollständig
 + hohe Kompression
 (bei höchstens gleichem Aufwand mindestens gleich hohe Mittelwerte und Minima wie Wörterbuch Kao und DMTF)

Gliederung

1. Einleitung

2. Stand der Technik

3. Modellierung der On-Chip-Trace-Architektur

4. Ergebnisse

5. Zusammenfassung

Modellierung der On-Chip-Trace-Architektur

Systematik der Modellierung - Befehlstypen

Befehlstyp	sequentiell	Verzweigung direkt	Verzweigung indirekt
unbedingt	zählen	zählen	Adresse
bedingt, ausgeführt	Ausführungsbit = "1"	Ausführungsbit = "1"	Adresse
bedingt, nicht ausgeführt	Ausführungsbit = "0"	Ausführungsbit = "0"	Ausführungsbit = "0"

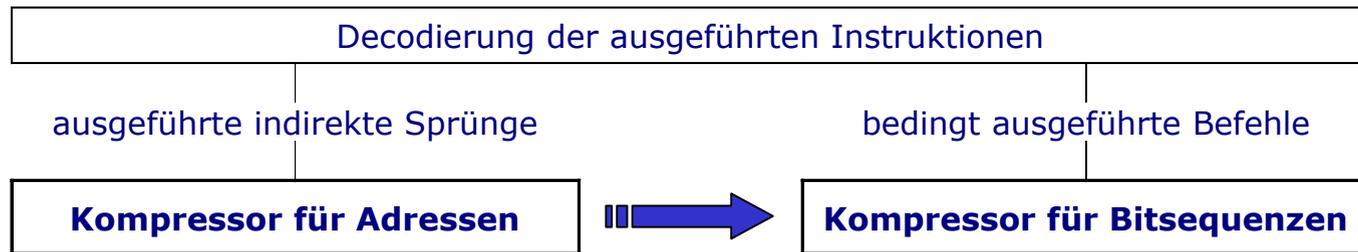
- aus Sicht des Trace nur drei Arten von Befehlen:
 - ausgeführte indirekte Sprünge: erzeugen eine **Adresse** in Adressbus-Breite
 - bedingte Befehle (außer bedingte ausgeführte indirekte Sprünge): erzeugen genau ein **Ausführungsbit**, das in einer **Bitsequenz** aufgezeichnet wird
 - alle restlichen Befehle: zählen (kleiner Anteil am Datenvolumen)
- Ermittlung der Anteile am abgeschätzten Volumen der Ausgabedaten:
 - bei einigen Programmen ist Anteil der **Adressen** nahezu 100 %
 - bei einigen Programmen ist Anteil der **Ausführungsbits** 100 %

Modellierung der On-Chip-Trace-Architektur

Systematik der Modellierung - Entwurfsentscheidungen

- große statistische Variabilität und Verschiedenartigkeit der Anteile (Adressen fester Breite versus Bitsequenzen beliebiger Längen) =>

Entwurfsentscheidung 1: separate Kompressoren für Adressen und für Bitsequenzen aus Ausführungsbits
= **architektonisches Novum**



- zeitliche Relation zwischen den Kompressoren muss gewahrt sein
- im Mittel folgt auf 48.000 bedingte Befehle ein indirekter Sprung =>

Entwurfsentscheidung 2: ein ausgeführter indirekter Sprung erzwingt die Ausgabe der aktuellen Bitsequenz

Modellierung der On-Chip-Trace-Architektur

Systematik der Modellierung - Kompressoren

- Konsequenz: **Kompressionspotential** bleibt **ungenutzt**
 - Wiederholungen von Sequenzen von Adressen
 - Wiederholungen von Abfolgen von Adressen und Bitsequenzen
- Kompensation: optionaler Kompressor für Messages
komprimiert Message-Sequenzen



Modellierung der On-Chip-Trace-Architektur

Kompressor für Adressen

Ausgangsbasis: mehrere existierende Lösungen aus der Literatur

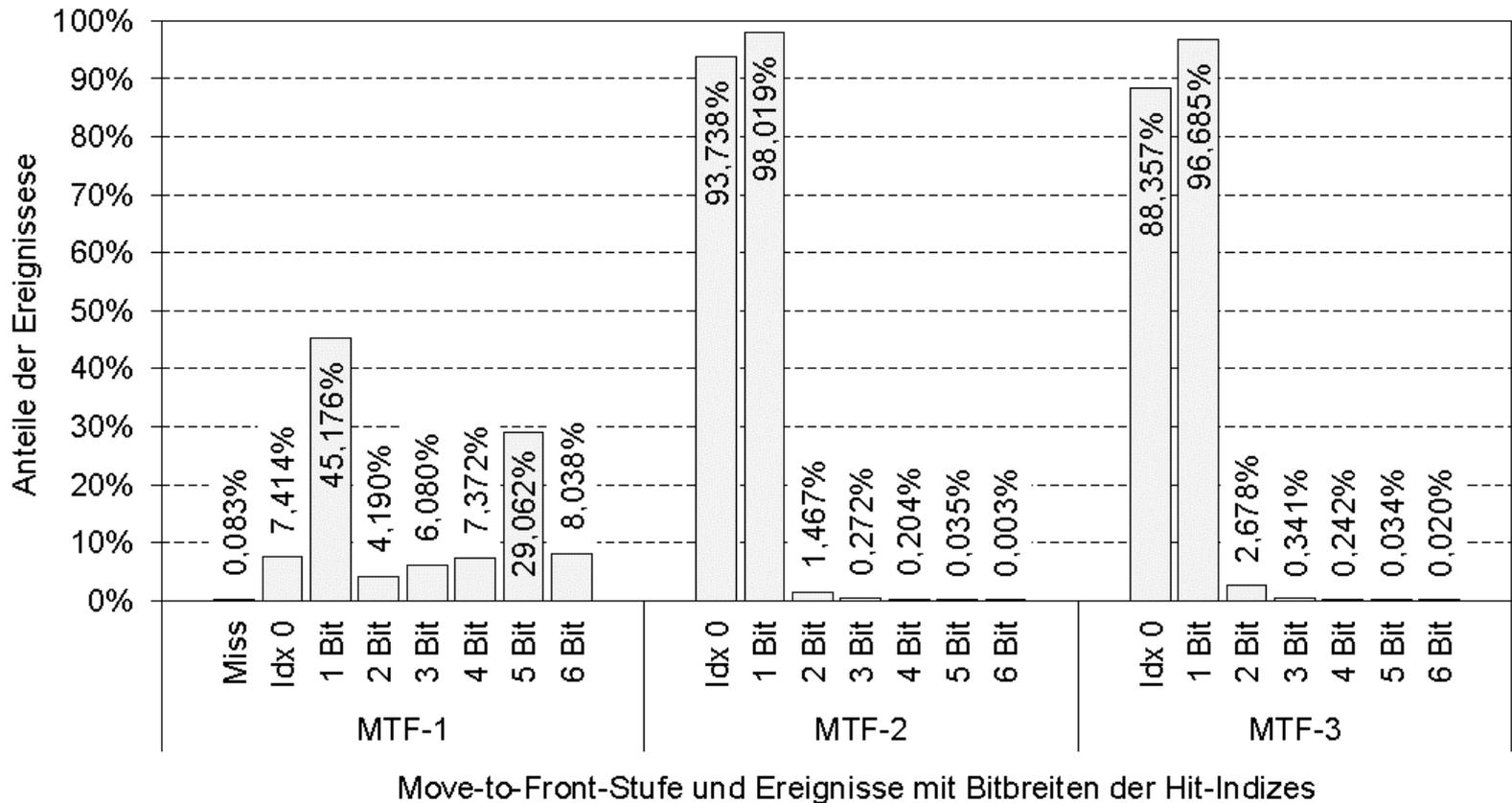
Wörterbuchansatz von Kao et al.	Double-Move-to-Front (DMTF) von Uzelac et al.
+ höherer Median des Kompressionsfaktors	+ höheres Minimum des Kompressionsfaktors (bessere Kompression kritischer Programme)
- Verdrängung unikater Werte	+ keine Verdrängung unikater Werte - feste Bitbreite der Längenangabe im Instruktionsstrom

=> Entscheidung: **mehrstufiger Move-to-Front-Ansatz**

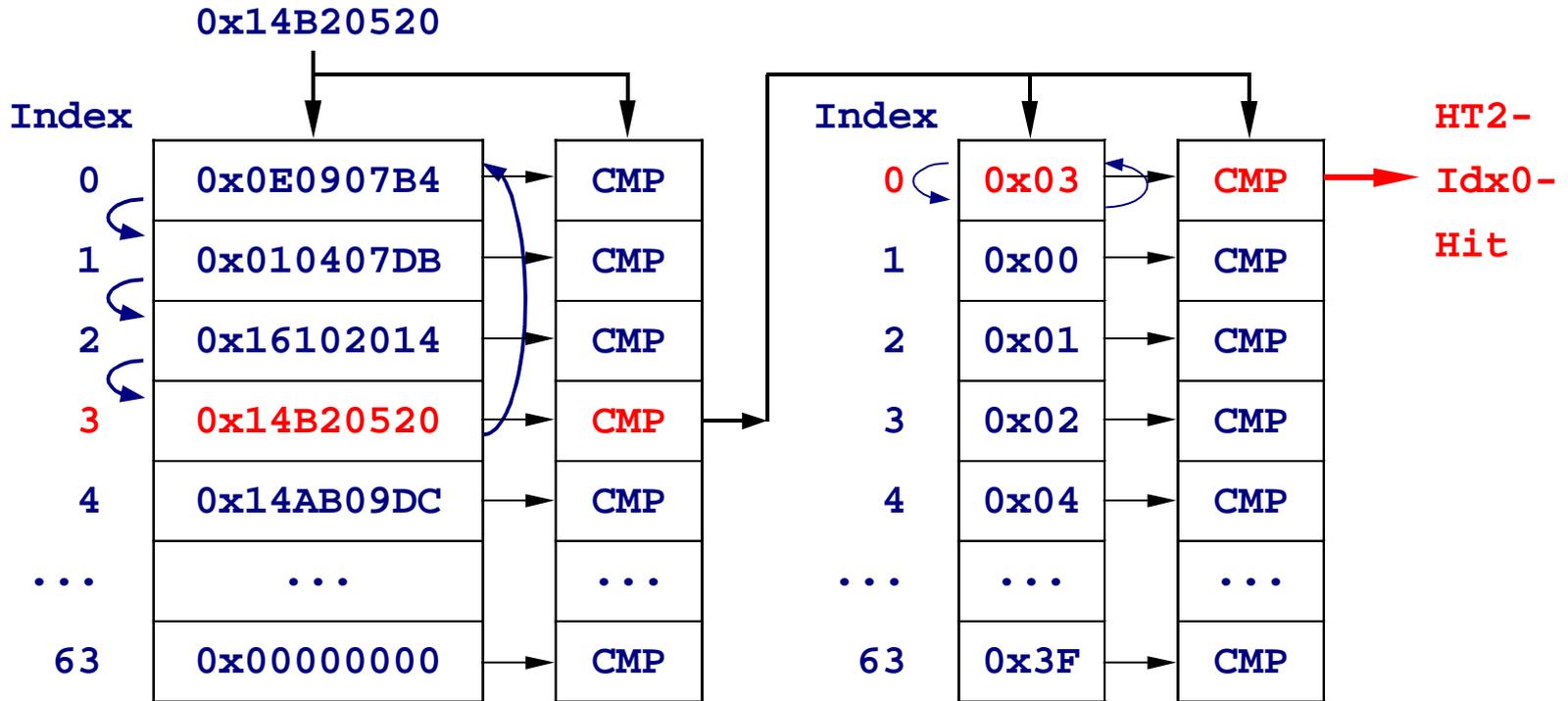
- **alle Stufen** haben die **gleiche Länge**
(64 Einträge => Indexbreite = 6 Bit)
- **Folgestufen** sind mit den eigenen Indizes **vorgeladen**
(=> keine Miss-Ereignisse ab zweiter Stufe)

Modellierung der On-Chip-Trace-Architektur

Kompressor für Adressen - Verdichtung der Hit-Ereignisse



Modellierung der On-Chip-Trace-Architektur Kompressor für Adressen - Funktionsprinzip



Modellierung der On-Chip-Trace-Architektur

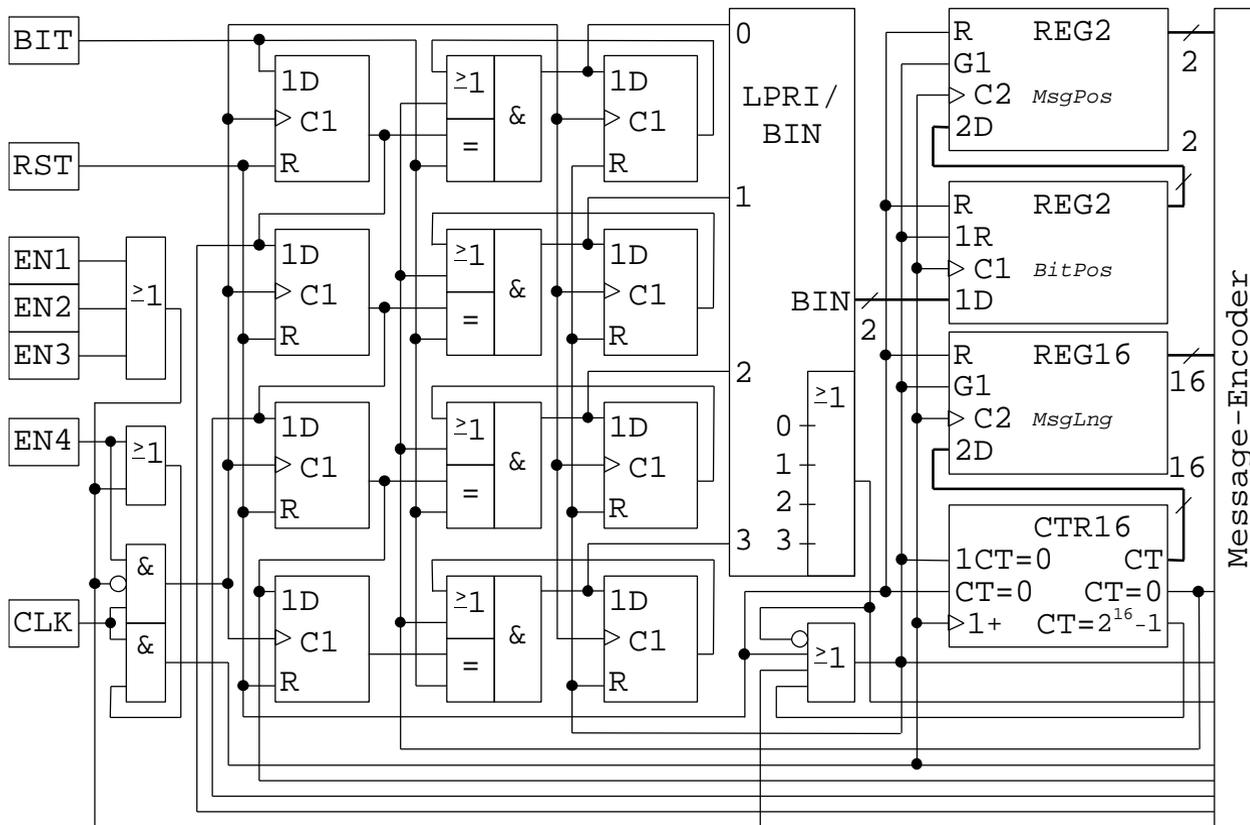
Kompressor für Bitsequenzen

- die Kompression von Bitsequenzen beliebiger Längen direkt in Hardware ist ein weitestgehend unbehandeltes Feld
- Analyse typischer Eigenschaften der Bitsequenzen
 - theoretisch 100 %; im Mittel 14,5 %, maximal 45,8 % bedingte Befehle
 - Sequenzlängen im Bereich 2...15 sind am häufigsten
 - Sequenzlängen ab 256 treten selten auf
 - Verteilung von Nullen und Einsen im Mittel weitgehend ausgewogen
 - Kompressionspotential in konsekutiven Folgen von Nullen oder Einsen durch Zählung wird als zu gering bewertet
- MTF-Ansätze sind ungeeignet
 - nur zwei unikate Werte => Tabellen haben nur zwei Einträge
 - Clusterbildung ungünstig für Detektion von Wiederholungen auf Bitebene

=> Entscheidung: **gleitendes Wörterbuch für Einzelbits**

Modellierung der On-Chip-Trace-Architektur

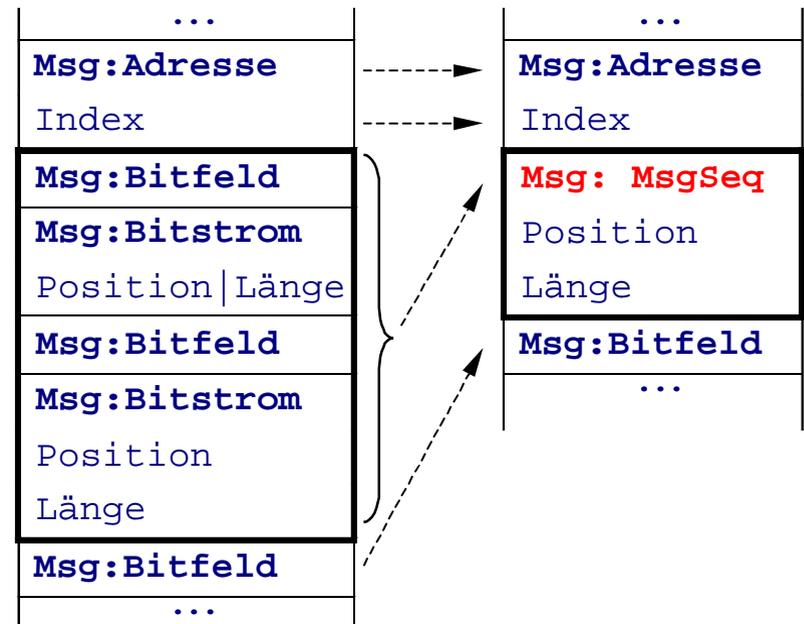
Kompressor für Bitsequenzen - Prinzipschaltung



Modellierung der On-Chip-Trace-Architektur

Kompressor für Messages

- Ziel: Kompression sich wiederholender Sequenzen aus Message-Bytes
- Signalisationsproblem: kein unikater Headercode für eine Sequenz-Message auf Byte-Ebene (im Payload kann jedes Bitmuster auftreten)
- **Lösung: Message-Ebene statt Byte-Ebene**
 - Messages werden entweder vollständig oder gar nicht komprimiert
 - eine Sequenz-Message kann nur an Positionen auftreten, an denen eine neue Message erwartet wird
=> unikater 1-Byte-Header ist wieder möglich



Modellierung der On-Chip-Trace-Architektur

Kompressor für Messages

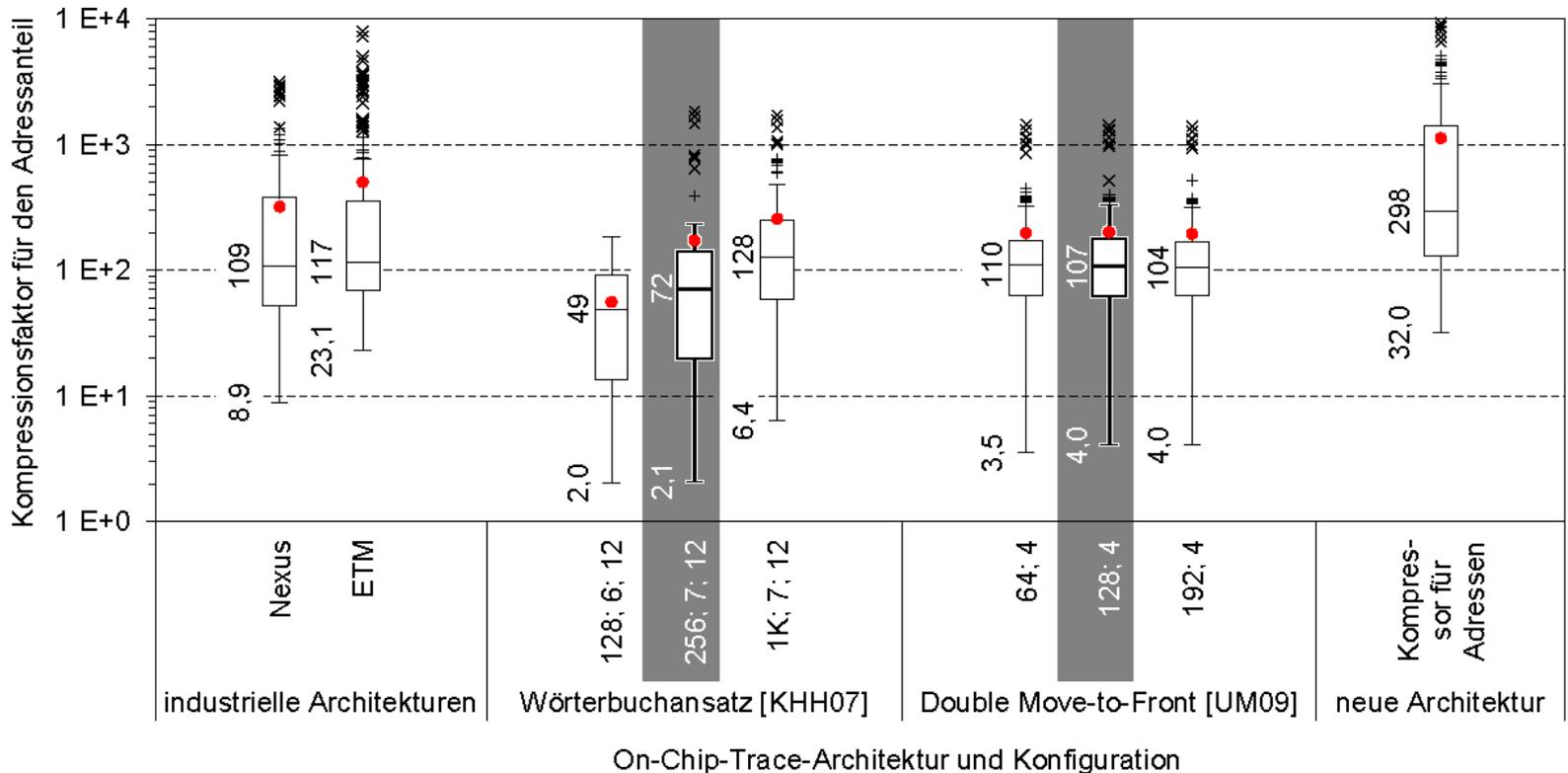
- MTF-Ansätze sind ungeeignet
 - Zerstörung der zeitlichen Relation
 - Barrel-Shifter für Message-Rotation
- => Entscheidung: **Wörterbuch für Messages**
- beliebig lange Messages werden als je ein Wörterbucheintrag behandelt
 - vereinfachte **Message-Decoder** für Detektion des ersten und letzten Bytes

Gliederung

1. Einleitung
2. Stand der Technik
3. Modellierung der On-Chip-Trace-Architektur
- 4. Ergebnisse**
5. Zusammenfassung

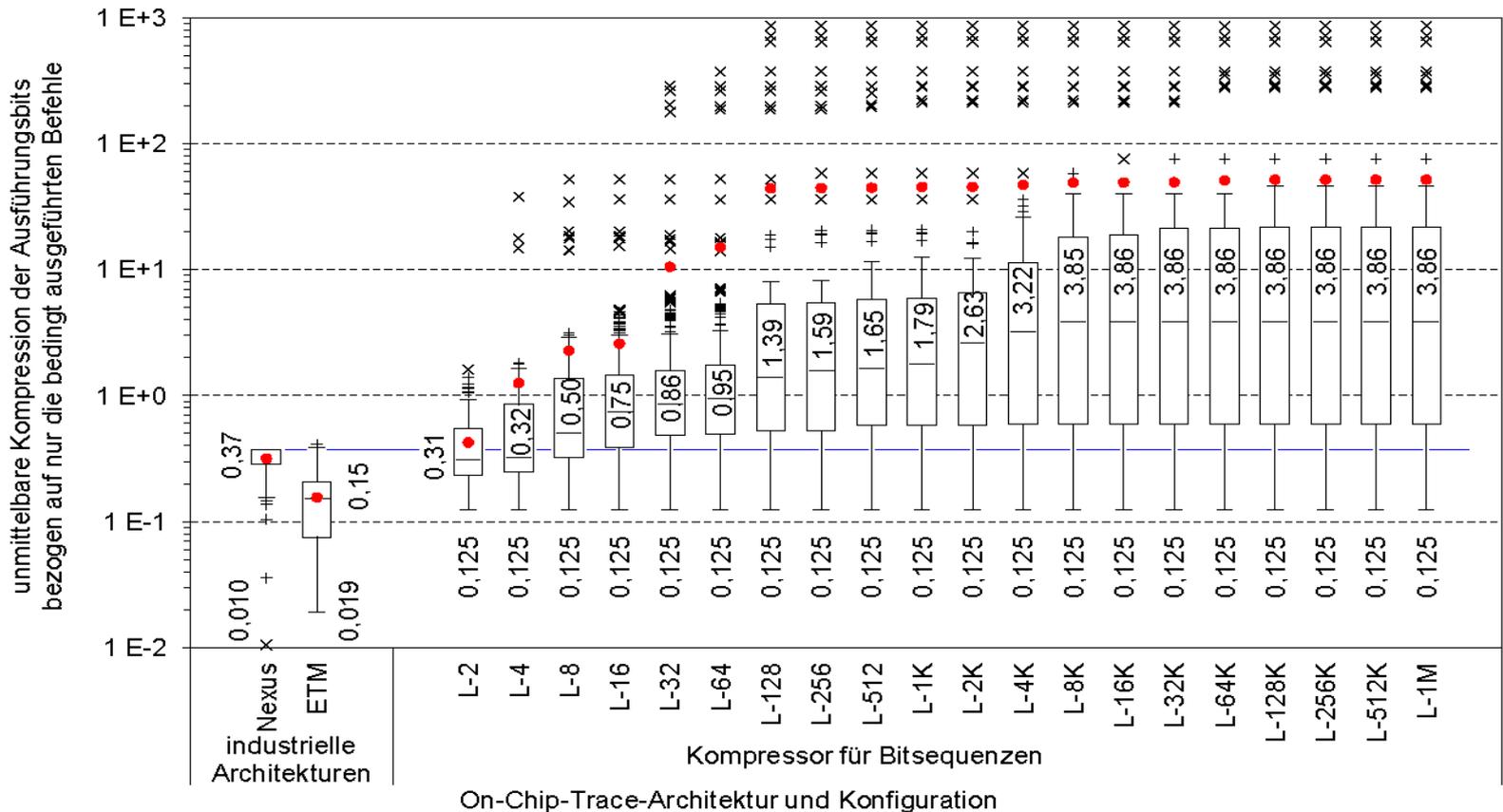
Ergebnisse

Kompressor für Adressen - Kompressionsfaktor



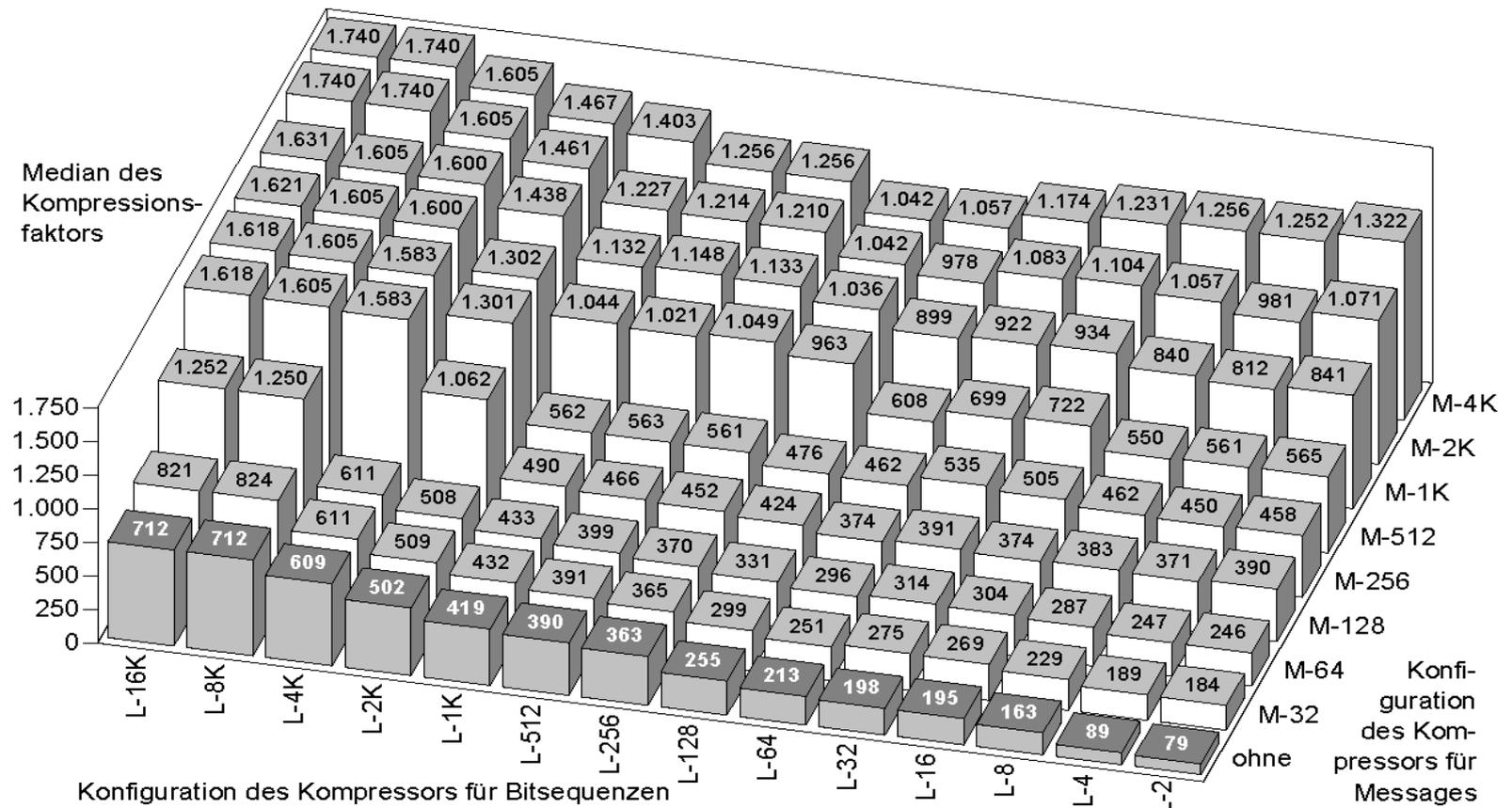
Ergebnisse

Kompressor für Bitsequenzen - unmittelbare Kompression



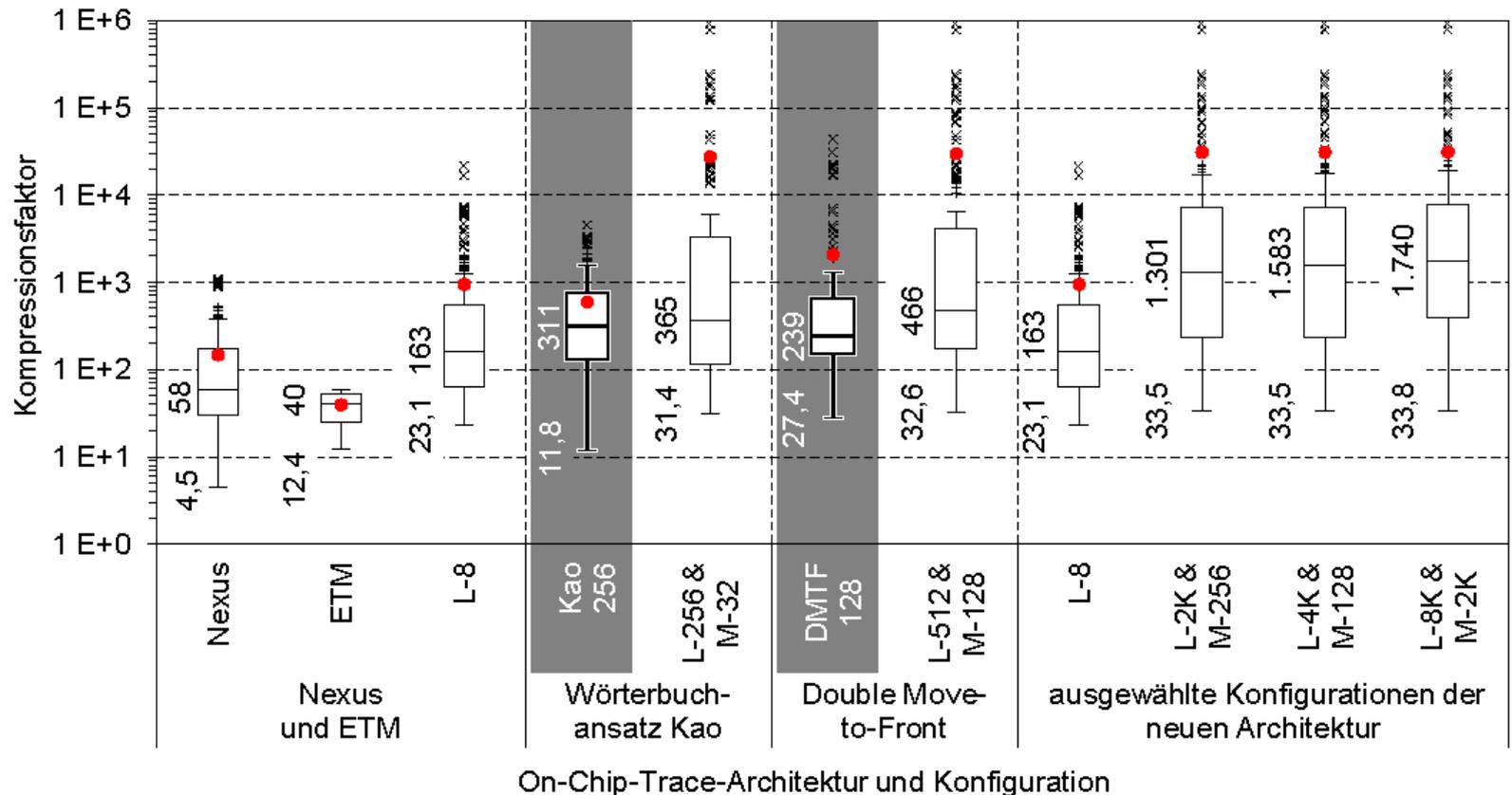
Ergebnisse

Gesamtsystem - Kompressionsfaktor



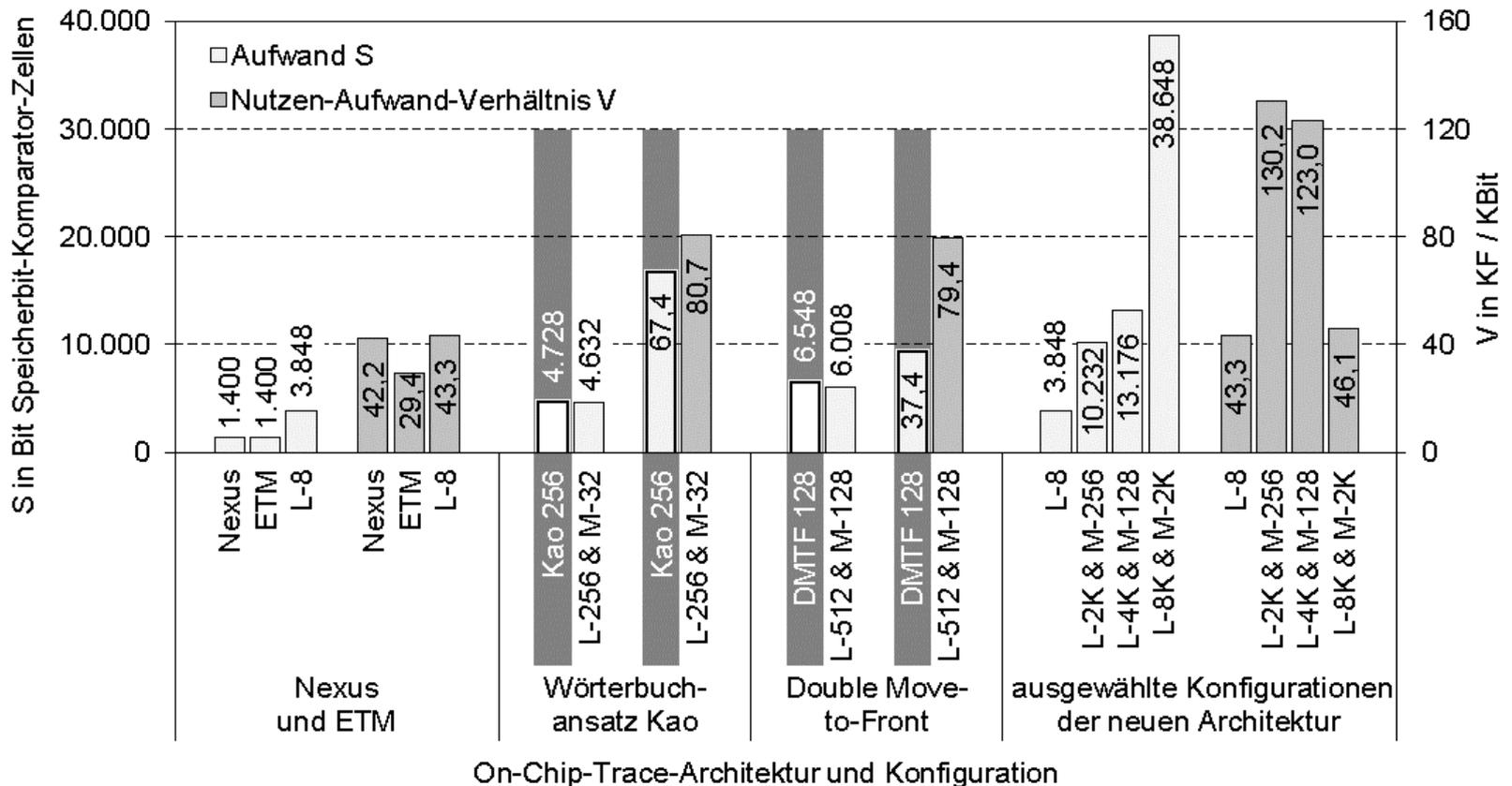
Ergebnisse

Vergleich: Kompressionsfaktor



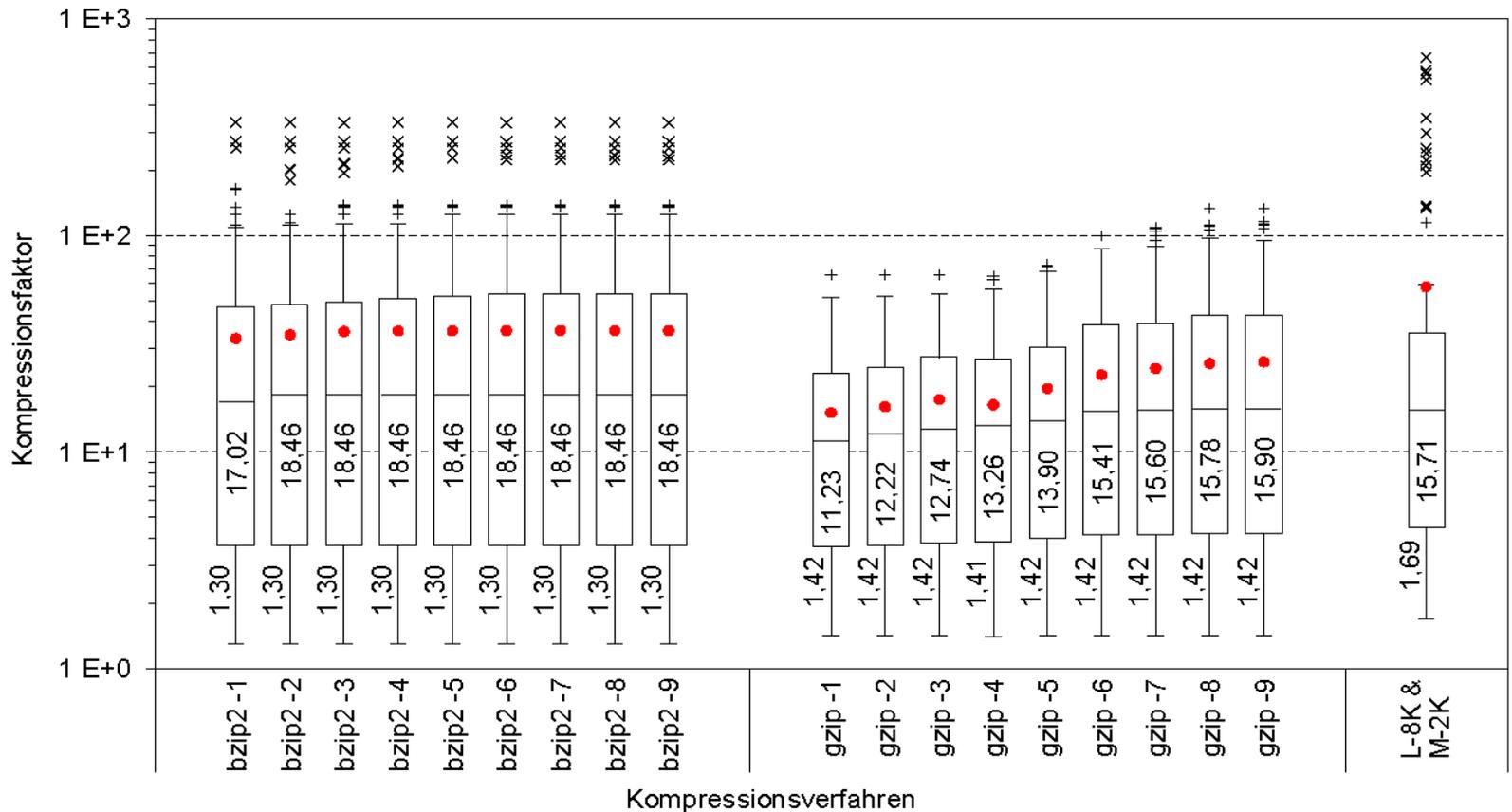
Ergebnisse

Vergleich: Aufwand und Nutzen-Aufwand-Verhältnis



Ergebnisse

Vergleich: Kompressionsfaktor gegenüber bzip2 und gzip



Ergebnisse

Implementierung

Kompressor für Bitsequenzen mit Wörterbuchlänge 256 Bit:

FPGA	Register	LUTs	Taktfrequenz
XC5VLX50T-FF1136-2 (speed)	550	1635	218 MHz
XC5VLX50T-FF1136-2 (size)	550	1428	107 MHz

Irrgang, K.-U. ; Preußner, T. B. ; Spallek, R. G.: An LZ77-Style Bit-Level Compression for Trace Data Compaction. *International Conference on Field Programmable Logic and Applications (FPL)*, 2015 (akzeptiert)

Irrgang, K.-U. ; Preußner, T. B. ; Spallek, R. G.: Kompression von Tracedaten auf der Basis eines auf Bitebene arbeitenden LZ77-Wörterbuchansatzes. *Fehlertolerante und energieeffiziente eingebettete Systeme: Methoden und Anwendungen (FEES)*, 2015 (akzeptiert)

Gliederung

1. Einleitung

2. Stand der Technik

3. Modellierung der On-Chip-Trace-Architektur

4. Ergebnisse

5. Zusammenfassung

Zusammenfassung und Ausblick

- **On-Chip-Trace:** Volumenreduktion unverzichtbar => Kompression
- kein **vollständiger Instruktionstrace bei hoher Kompression** bisher => Schließen dieser Lücke = **Hauptziel**
- Analyse typischer Benchmarkprogramme => Entwurfsentscheidungen
=> **Architektur: 2 + 1 Kompressoren**
 - Kompressor für Adressen: zweistufiges Move-to-Front
 - Kompressor für Bitsequenzen: Wörterbuch für Einzelbits
 - Kompressor für Messages: Wörterbuch für Messages
- **Ziele wurden erreicht** => Überlegenheit der neuen Architektur

Ausblick:

- geeignetes Vorladen der Kompressoren (und Dekompressoren)
- Optimierung der Message-Codierung
- Eignung der Kompressionsansätze für Datentrace u. a.

Vielen Dank
für Ihre
Aufmerksamkeit!